

## Метод адаптивных взвешенных сумм в задаче Парето-аппроксимации

# 06, июнь 2012

DOI: 10.7463/0612.0423283

Карпенко А. П., Савелов А. С., Семенихин А. С.

УДК 519.6

Россия, МГТУ им. Н.Э. Баумана

[apkarpenko@mail.ru](mailto:apkarpenko@mail.ru)

[saspost@yandex.ru](mailto:saspost@yandex.ru)

### Введение

Постановка задачи многокритериальной оптимизации (МКО-задачи) фиксирует множество допустимых значений вектора варьируемых параметров задачи и вектор критериальных функций. Данная информация позволяет обычно выделить не одно решение задачи, а лишь ее множество Парето. Поэтому часто говорят, что решением МКО-задачи является множество Парето этой задачи. Множество Парето и фронт Парето занимают в теории многокритериальной оптимизации исключительное место еще и потому, что согласно известному принципу Эджворта-Парето, при «разумном» поведении лица, принимающего решения (ЛПР), выбор решения следует производить на множестве Парето [1].

Классические методы решения МКО-задачи основаны на использовании, помимо указанной информации о задаче, еще тем или иным образом формализованной информации о предпочтениях ЛПР. В результате задачу удается свести к совокупности задач глобальной однокритериальной оптимизации. Относительно новый и быстро развивающийся класс методов решения МКО-задачи образуют *методы Парето-аппроксимации*,

предполагающие предварительное построение некоторой конечномерной аппроксимации множества, а тем самым, и фронта Парето.

Известно большое число популяционных и непопуляционных методов построения Парето-аппроксимации (см., например, обзор [2]). Работа посвящена исследованию эффективности, так называемого, метода адаптивных взвешенных сумм (*Adaptive Weighted Sum method, AWS-method*), который предложили и разработали Рю, Ким и Ван (*J.-H. Ryu, S. Kim, H. Wan*) [3]. Для решения задачи Парето-аппроксимации метод *AWS* использует аддитивную свертку частных критериев оптимальности. Однако в отличие от классического метода суммы взвешенных критериев (*Weighted Sum method, WS-method*) [1], также использующего такую свертку, метод *AWS* предполагает адаптацию весовых коэффициентов в процессе итераций на основе информации о текущем положении подобласти поиска. Целью разработки метода *AWS* было преодоление известного недостатка метода *WS*, заключающегося в невозможности локализации точек множества Парето, которые соответствуют вогнутым фрагментам фронта Парето.

Для сокращения затрат на вычисление значений критериальных функций метод *AWS* использует метамоделли этих функций. Поскольку в практически значимых задачах вычислительная сложность критериальных функций обычно является высокой, это обстоятельство позволяет сократить вычислительные затраты и делает метод, на наш взгляд, весьма перспективным. Недостатком метода *AWS* в существующем виде является его ориентация на решение только двухкритериальных задач Парето-аппроксимации.

Выполненное авторами метода *AWS* исследование показало его высокую эффективность. Однако исследование было выполнено всего для двух относительно простых тестовых задач многокритериальной оптимизации [3]. В данной работе мы повторяем исследование для указанных тестовых задач, а также дополнительно проводим исследование для трех других сложных тестовых задач.

В содержательных терминах качество Парето-аппроксимации может быть оценено с помощью следующих характеристик:

- близость найденных решений к точному множеству Парето рассматриваемой МКО-задачи;
- равномерность распределения решений в полученной Парето-аппроксимации;
- мощность найденного множества решений.

Известно большое число индикаторов качества, формализующих указанные характеристики [4, 5]. В данной работе в качестве индикатора качества Парето-аппроксимации используем мощность найденной аппроксимации.

В первом разделе даем постановку МКО-задачи и схему метода *AWS*. Во втором разделе рассматриваем программную реализацию метода и организация экспериментов. В третьем разделе представлены результаты экспериментов и их обсуждение. В заключении сформулированы основные выводы и перспективы развития работы.

Научная новизна работы заключается в исследовании эффективности Парето-аппроксимации методом *AWS* на тестовых задачах, на которых ранее его эффективность не исследовалась. Результаты исследования выявили ряд ограничений метода и сложностей его использования. Это позволило наметить пути модификации метода, направленные на преодоление этих ограничений и сложностей.

## 1. Постановка задачи и схема метода

Пусть множеством допустимых значений вектора варьируемых параметров  $X$  является ограниченное и замкнутое множество  $D_X \subset \{X\} = R^{|X|}$ . Положим, что критериальная вектор-функция  $F(X) = (f_1(X), f_2(X), \dots, f_{|F|}(X))$  со значениями в критериальном пространстве  $\{F\} = R^{|F|}$  определена в области  $D_X$ . ЛПР стремится

минимизировать в этой области каждый из частных критериев оптимальности  $f_1(X), f_2(X), \dots, f_{|F|}(X)$ , что условно записываем в виде

$$\min_{X \in D_X} F(X) = F(X^*) = F^*, \quad (1)$$

где векторы  $X^*, F^*$  - искомое решение задачи многокритериальной оптимизации.

Здесь и далее запись вида  $|A|$ , где  $A$  - вектор или некоторое счетное множество, означает размерность этого вектора или мощность множества соответственно. Полагаем, что частные критерии оптимальности тем или иным образом нормализованы, так что  $f_i(X) \in [0; 1]$  для любого  $X \in D_X$ ;  $i \in [1: |F|]$ .

Вектор-функция  $F(X)$  выполняет отображение множества  $D_X$  во множество  $D_F \in \{F\}$ , которое называется множеством достижимости. Множество и фронт Парето обозначаем  $D_X^* \subset D_X$ ,  $D_F^* \subset D_F$ , а их конечномерные аппроксимации  $\Theta^X$ ,  $\Theta^F$  соответственно. Последние множества называем архивными.

Метод AWS включает в себя три следующие основные процедуры:

- определение центральной точки;
- формирование метамоделей частных критериев оптимальности;
- решение полученных оптимизационных задач.

Рассмотрим суть указанных процедур.

*Определение центральной точки.* На этапе инициализации центральную точку  $X_C^0$  выбираем случайным образом в области  $D_X$ . На этом же этапе должны быть определены следующие свободные параметры алгоритма:  $\delta$  - начальный радиус области доверия (*trust region radius*);  $\rho \in (0; 1)$  - коэффициент сужения этой области;  $\delta_{\min}$  - минимальная величина радиуса области.

На итерации  $(t + 1)$  центральную точку  $X'_C$  отыскиваем среди точек текущей Парето-аппроксимации  $\Theta^X = \Theta^X(t)$ , построенной на предыдущей итерации  $t$ .

Отсортируем элементы архивных множеств  $\Theta^X$ ,  $\Theta^F$  по возрастанию первого частного критерия  $f_1(X)$  и представим в виде линейных списков с прежними наименованиями. Определим расстояние  $d_j$  архивной точки  $F_j^\Theta$  до ближайших к ней в списке  $\Theta^F$  точек формулой

$$d_j = \|F_{j-1}^\Theta - F_j^\Theta\|_E + \|F_j^\Theta - F_{j+1}^\Theta\|_E, \quad j \in [1:|\Theta|], \quad (3)$$

где  $\|\bullet\|_E$  - евклидова норма.

Алгоритм использует следующее правило определения центральной точки  $X'_C$ .

1) Если  $|\Theta| > 2$ , то полагаем  $X'_C = X_{j^*}^\Theta$ , где

$$j^* = \arg \left( \max_{j \in [2:|\Theta|-1]} d_j \mid X_j^\Theta \notin X_T \right).$$

Здесь  $X_T$  - множество точек, использованных в качестве центральных на всех предыдущих итерациях  $[0: t]$ . Иными словами, за центральную точку принимаем точку, во-первых, наиболее удаленную от других точек множества  $\Theta^X$  в смысле расстояния (3), и, во-вторых, не использованную на предшествующих итерациях.

2) Если  $|\Theta| = 2$ , то с равной вероятностью полагаем  $X'_C = X_1^\Theta$  или  $X'_C = X_2^\Theta$ .

3) Если  $|\Theta| = 1$ , то принимаем  $X'_C = X_1^\Theta$ .

*Формирование метамоделей.* Метамоделю представляют собой квадратичные аппроксимации  $m'_1(X)$ ,  $m'_2(X)$  функций  $f_1(X)$ ,  $f_2(X)$  в окрестности точки  $X'_C$ :

$$m'_1(X) = f_1(X'_C) + \nabla^T f_1(X'_C)(X - X'_C) + \frac{1}{2}(X - X'_C)^T H_1(X'_C)(X - X'_C);$$

$$m'_2(X) = f_2(X'_C) + \nabla^T f_2(X'_C)(X - X'_C) + \frac{1}{2}(X - X'_C)^T H_2(X'_C)(X - X'_C).$$

Здесь  $H_1(X'_C)$ ,  $H_2(X'_C)$  - матрицы Гессе функций  $f_1(X)$ ,  $f_2(X)$  в точке  $X'_C$ .

Если  $|\Theta| > 2$ , то дополнительно строим метамоделли

$$m'_p(X) = \lambda_1^p m'_1(X) + \lambda_2^p m'_2(X),$$

$$m'_q(X) = \lambda_1^q m'_1(X) + \lambda_2^q m'_2(X),$$

а если  $|\Theta| = 2$  или  $|\Theta| = 1$  - метамоделль

$$m'_p(X) = \lambda_1^p m'_1(X) + \lambda_2^p m'_2(X).$$

В первом случае ( $|\Theta| > 2$ ) весовые множители  $(\lambda_1^p, \lambda_2^p) = \Lambda^p$ ,  $(\lambda_1^q, \lambda_2^q) = \Lambda^q$  определяем по правилу

$$\Lambda^p = c^p \left( (-f_2(X'_C) + f_2(X_{j^*-1}^\ominus)), (f_1(X'_C) - f_1(X_{j^*-1}^\ominus)) \right),$$

$$\Lambda^q = c^q \left( (-f_2(X_{j^*+1}^\ominus) + f_2(X'_C)), (f_1(X_{j^*+1}^\ominus) - f_1(X'_C)) \right);$$

во втором случае ( $|\Theta| = 2$ ) – по правилу

$$\Lambda^p = c^p \left( (-f_2(X_2^\ominus) + f_2(X_1^\ominus)), (f_1(X_2^\ominus) - f_1(X_1^\ominus)) \right);$$

в третьем случае (когда  $|\Theta| = 1$ ) – по правилу  $\Lambda^p = (0,5, 0,5)$ . Константы  $c^p$ ,  $c^q$  выбираем таким образом, чтобы обеспечить выполнение условий нормировки  $\lambda_1^p + \lambda_2^p = \lambda_1^q + \lambda_2^q = 1$ .

Отметим, что при построении метамоделей  $m'_1(X)$ ,  $m'_2(X)$  речь может идти не о градиентах и матрице Гессе функций  $f_1(X)$ ,  $f_2(X)$ , а об их оценках, полученных, например, численными методами (путем соответствующих конечно-разностных аппроксимаций указанных функций).

*Решение оптимизационных задач.* Данная процедура предполагает решение задач оптимизации

$$\min_{X \in D'_C} m'_1(X) = m'_1(\hat{X}'_1), \quad \min_{X \in D'_C} m'_2(X) = m'_2(\hat{X}'_2), \quad (4)$$

где текущую область доверия (trust region)  $D'_C$  определяет формула

$$D'_C = \{X \mid X \in D_X, \|X - X'_C\| \leq \delta\}.$$

Если  $|\Theta| > 2$ , то решения  $\hat{X}'_1, \hat{X}'_2$  задач (4) позволяют отыскать приближенно оптимальные по Парето точки  $\hat{X}'_p, \hat{X}'_q$ , принадлежащие области доверия  $D'_C$ , путем решения оптимизационных задач

$$\min_{X \in D'_C} m'_p(X) = m'_p(\hat{X}'_p), \quad \min_{X \in D'_C} m'_q(X) = m'_q(\hat{X}'_q). \quad (5)$$

Данный этап алгоритма иллюстрирует рисунок 1, на котором принято что  $F'_C = F(X'_C)$ ,  $F_{j^*-1}^\ominus = F(X_{j^*-1}^\ominus)$ ,  $F_{j^*+1}^\ominus = F(X_{j^*+1}^\ominus)$ .

Отметим, что задачи (4), (5) представляют собой задачи оптимизации квадратичных функций, для решения которых известны высокоэффективные методы, алгоритмы и соответствующее программное обеспечение.

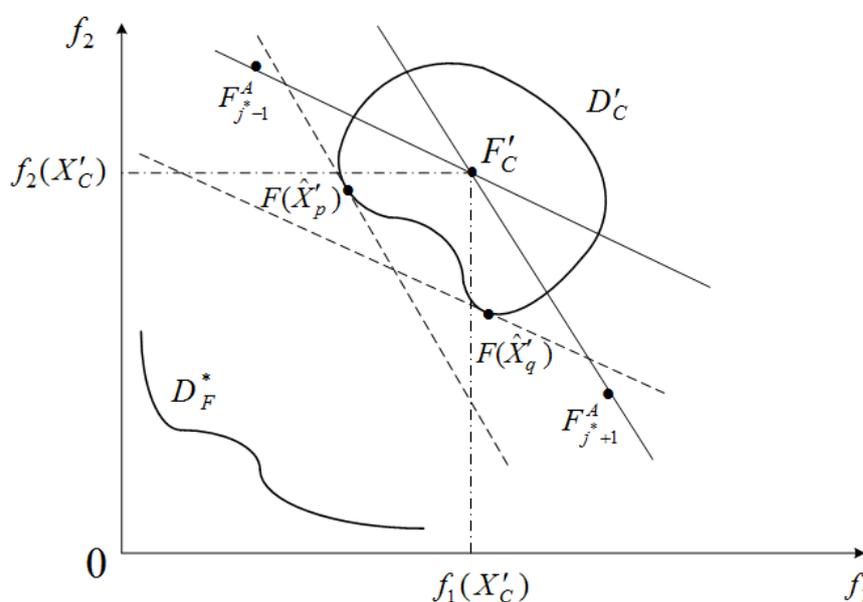


Рисунок 1 – К схеме метода AWS: результаты решения задач (5)

В процессе итераций текущий радиус области доверия уменьшают по правилу  $\delta = \rho \delta$  до достижения минимально допустимой его величины  $\delta_{\min}$ .

Новое состояние архивного множества  $\Theta^X$  получаем путем добавления в него точек  $\hat{X}'_1, \hat{X}'_2, \hat{X}'_p, \hat{X}'_q$  и исключения из полученного набора доминируемых решений. Аналогично, множество  $\Theta^F$  образуем путем добавления в него точек  $F(\hat{X}'_1), F(\hat{X}'_2), F(\hat{X}'_p), F(\hat{X}'_q)$ .

## 2. Программная реализация метода и организация экспериментов

Для решения однокритериальных задач условной оптимизации (4), (5) в программной реализации использован метод штрафных функций в совокупности с известным методом Нелдера-Мида локальной безусловной оптимизации. Вместо метамоделей критериальных функций  $f_1(X), f_2(X)$  использованы сами эти функции. Данное ограничение не является критичным, поскольку влияет на время вычислений, но не на качество Парето-аппроксимации.

Программа написана на языке программирования C++. Ниже представлены основные классы программной модели:

- *AWSAlgorithm* – класс, реализующий алгоритм AWS;
- *PenaltyAlgorithm* – класс, реализующий метод штрафных функций;
- *NelderMidAlgorithm* – класс, реализующий алгоритм Нелдера-Мида;
- *Polyhedron* – вспомогательный класс, реализующий основные операции над многогранниками;
- *QuickSort* – класс, реализующий алгоритм «быстрой сортировки».
- *Function* – абстрактный класс, реализующий общий интерфейс для реализаций критериальных функций и ограничений;
- *ZDT3, CompositeFunction* – конкретные реализации критериальных функций;
- *HyperSphereConstrain* - конкретная реализация ограничений.

Эксперименты выполнены на персональном компьютере, основные параметры которого представлены ниже:

- процессор Intel Pentium Dual-Core T4500, 2,3 ГГц;

— оперативная память DDR-3, 3 Гб.

Изучалось влияние на эффективность метода следующих его свободных параметров:

- начальный радиус области доверия  $\delta$ ;
- коэффициент сужения области доверия  $\rho$ ;
- минимальная величина радиуса области доверия  $\delta_{\min}$ .

В качестве меры эффективности метода использована *мощность множества решений* (*overall non-dominated vector generation*)

$$I_{ONVG}(\Theta) = |\Theta| \rightarrow \max,$$

то есть число элементов множества  $\Theta$  [5].

Исследование выполнено на пяти следующих известных двухкритериальных тестовых задачах многокритериальной оптимизации.

*Задача «о двух параболоидах»* (двумерная, двухкритериальная):

$$D_X = \{X / 0 \leq x_i \leq 1, i = 1, 2\};$$

$$f_1(x) = x_1^2 + x_2^2,$$

$$f_2(x) = (x_1 - 1)^2 + (x_2 - 1)^2.$$

Задача имеет непрерывный выпуклый фронт Парето.

*Задача Аудета* (C. Audet):

$$D_X = \{X / 0 \leq x_i \leq 1, i = 1, 2\};$$

$$f_1(X) = f_1(x_1) = 4 \cdot x_1,$$

$$f_2(X) = g(X) \cdot h(X),$$

где

$$h(X) = \begin{cases} 1 - \left( \frac{f_1(X)}{g(X)} \right)^\alpha, & f_1(X) \leq g(X), \\ 0, & \text{иначе,} \end{cases}$$

$$g(X) = g(x_2) = 4 - 3 \cdot \exp\left(-\left\{\frac{x_2 - 0,2}{0,02}\right\}^2\right).$$

Задача является двумерной, двухкритериальной, фронт Парето - непрерывный, значению  $\alpha = 0,25$  соответствует выпуклое множество Парето, а  $\alpha = 4$  - невыпуклое [6].

*Задача ZDT3* (30-мерная, двухкритериальная):

$$D_X = \{X / 0 \leq x_i \leq 1, i \in [1:|X|], |X| = 30\};$$

$$f_1(X) = f(x_1) = x_1,$$

$$f_2(X) = g(X) \cdot \left[ 1 - \sqrt{\frac{f_1(X)}{g(X)}} - \frac{f_1(X)}{g(X)} \cdot \sin(10 \pi f_1(X)) \right],$$

где

$$g(X) = g(x_2, \dots, x_{|X|}) = 1 + 9 \cdot \sum_{i=2}^{|X|} \frac{x_i}{|X| - 1}.$$

Сложность задачи обусловлена высокой размерностью, а также несвязным, хотя и выпуклым, фронтом Парето [4].

*Задача ZDT6* (10-мерная, двухкритериальная):

$$D_X = \{X / 0 \leq x_i \leq 1, i \in [1:|X|], |X| = 10\};$$

$$f_1(X) = f(x_1) = 1 - \exp(-4 x_1) \cdot \sin^6(6 \pi x_1),$$

$$f_2(X) = g(X) \cdot \left[ 1 - \left( \frac{f_1(X)}{g(X)} \right)^2 \right],$$

где

$$g(X) = g(x_2, \dots, x_{|X|}) = 1 + 9 \cdot \left( \sum_{i=2}^{|X|} \frac{x_i}{|X| - 1} \right)^{0,25}.$$

Задача имеет слабо невыпуклый фронт Парето [4]. Сложность задачи ZDT6 заключается в невыпуклости фронта и многомерности задачи.

*Задача ZDT7*: (30-мерная, двухкритериальная):

$$D_X = \{X / 0 \leq x_i \leq 1, i \in [1:|X|], |X| = 30\};$$

$$f_1(X) = f(x_1) = x_1,$$

$$f_2(X) = g(X) \cdot h(X),$$

где

$$g(X) = g(x_2, \dots, x_{|X|}) = 1 + 9 \cdot \sum_{i=2}^{|X|} \frac{x_i}{|X|-1},$$

$$h(X) = 1 - \sqrt{\frac{f_1(X)}{g(x_2, \dots, x_{|X|})}}.$$

Задача имеет непрерывный выпуклый фронт Парето [4].

### 3. Результаты экспериментов

*Задача «о двух параболоидах»* (рисунок 2). Результаты решения данной задачи, в силу ее простоты, показывают, по сути, только работоспособность алгоритма и программы *AWSM*, реализующей этот алгоритм. Из рисунка 2 следует, что за небольшое число итераций  $\hat{t} = 30$  построено множество  $\Theta$ , состоящее из  $I_{ONVG}(\Theta) = 105$  недоминируемых решений. Обеспечены приемлемая плотность и равномерностью покрытия фронта Парето. Результаты исследования показывают, что варьирование значений свободных параметров метода в широком диапазоне оказывает в данном случае слабое влияние на качество Парето-аппроксимации.

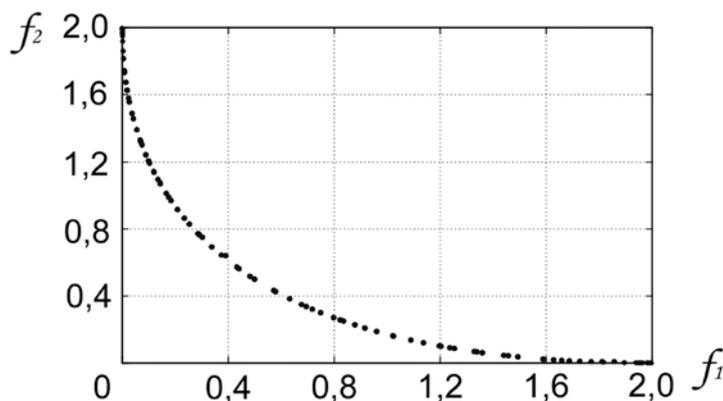


Рисунок 2 - Задача «о двух параболоидах»:  $\delta = 1$ ;  $\rho = 2$ ;  $\delta_{\min} = 0,001$ ;  $\hat{t} = 30$

*Задача Аудета.* (рисунки 3 - 6). Результаты решения задачи для случая  $\alpha = 0,25$ , когда фронт Парето является выпуклым, иллюстрируют рисунки 3, 4. Рисунок 3 показывает, что за  $\hat{t} = 30$  итераций удалось получить только  $I_{ONVG}(\Theta) = 26$  недоминируемых решений (против  $I_{ONVG}(\Theta) = 91$

решений, заявленных в исходной работе [3]). Метод не обеспечил равномерность покрытия фронта Парето. Заявленное число точек и приемлемую равномерность покрытия удалось достичь только при удвоенном числе итераций  $\hat{t} = 60$  (рисунок 4).

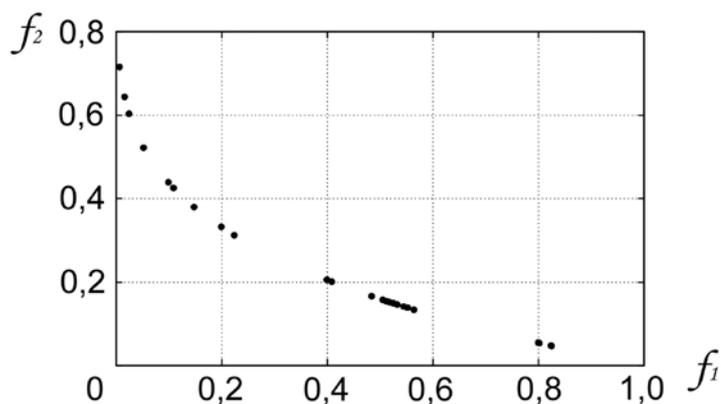


Рисунок 3 - Задача Аудета:  $\alpha = 0,25$ ;  $\delta = 0,2$ ;  $\rho = 2$ ,  $\delta_{\min} = 0,001$ ;  $\hat{t} = 30$

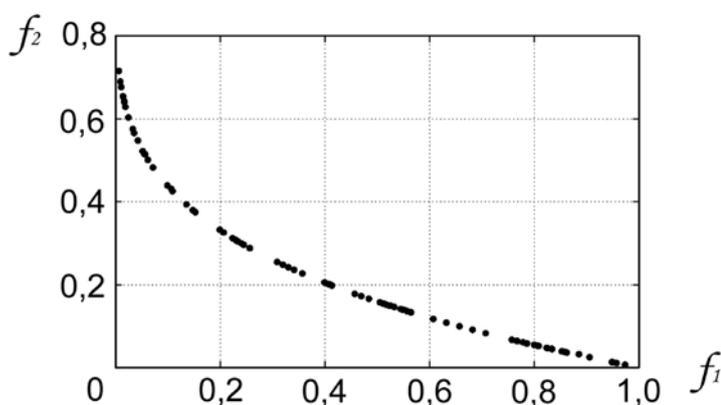


Рисунок 4 - Задача Аудета:  $\alpha = 0,25$ ;  $\delta = 0,2$ ;  $\rho = 2$ ,  $\delta_{\min} = 0,001$ ;  $\hat{t} = 60$

Возможная причина указанных недостатков метода заключается в использованной в программе *AWSM* модели планирования эксперимента, которая предполагает испытания только в осевых точках.

Результаты решения задачи Аудета для случая  $\alpha = 4$ , когда фронт Парето не является выпуклым, иллюстрируют рисунки 5, 6. За  $\hat{t} = 30$  итераций удалось получить только  $I_{ONVG}(\Theta) = 30$  решений (в работе [3] для

этого случая заявлено  $I_{ONVG}(\Theta) = 80$  решений). Заявленное число точек и приемлемую равномерность покрытия фронта Парето удалось достичь только при  $\hat{t} = 100$  итераций. Возможные причины отмеченных недостатков указаны выше.

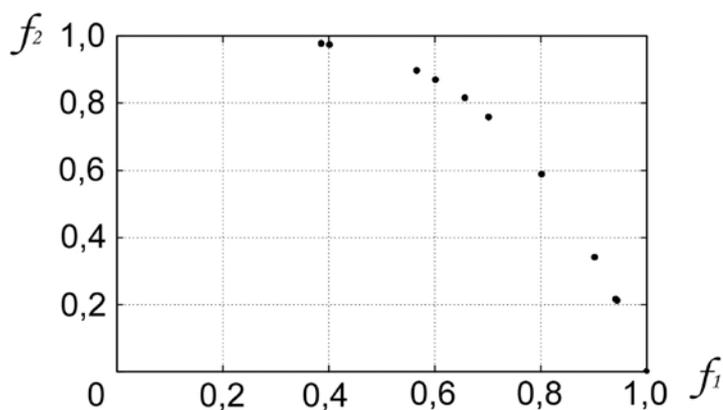


Рисунок 5 - Задача Аудета:  $\alpha = 4$ ;  $\delta = 0,2$ ;  $\rho = 2$ ,  $\delta_{\min} = 0,001$ ;  $\hat{t} = 30$

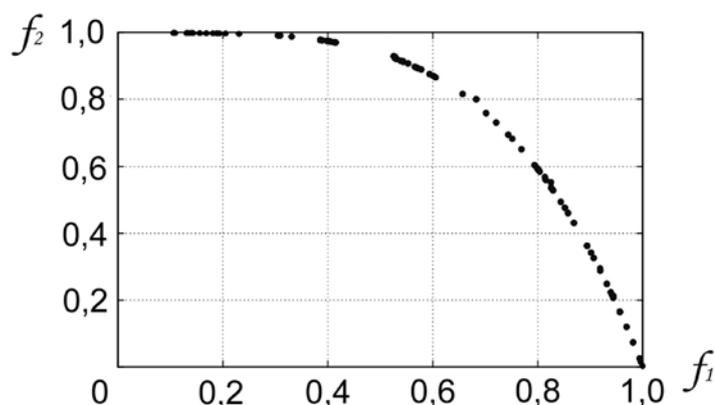


Рисунок 6 - Задача Аудета:  $\alpha = 4$ ;  $\delta = 0,2$ ;  $\rho = 2$ ,  $\delta_{\min} = 0,001$ ;  $\hat{t} = 100$

Тщательный анализ результатов, представленных на рисунке 6, показывает, что некоторое число полученных решений не удовлетворяет ограничениям задачи. Причина появления этих решений заключается в том, что задачи глобальной условной оптимизации (4), (5) решаются в программе *AWSM*, как мы указывали выше, методом штрафных функций. Априорное назначение требуемой точности удовлетворения ограничений затруднительно. В то же время, как мы видим, даже кажущаяся вполне

удовлетворительной точность 0,001 приводит к появлению в архиве  $\Theta$  непаретовских решений.

*Задача ZDT3* (рисунки 7, 8). Исследование показало, что качество полученной Парето-аппроксимации в данном случае сильно зависит от значений свободных параметров метода.

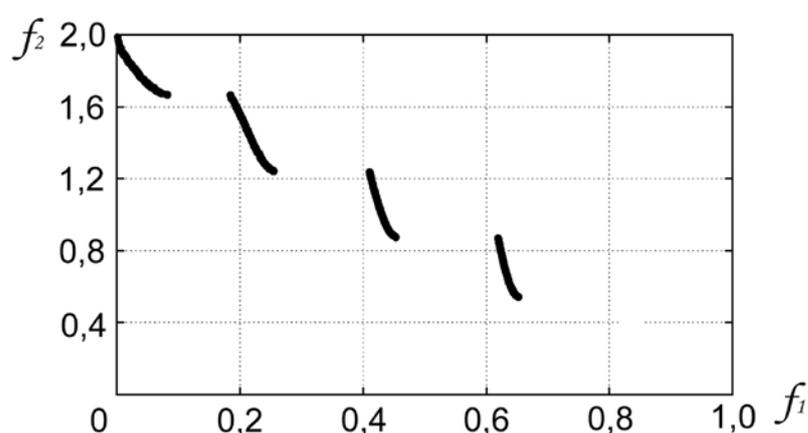


Рисунок 7 - Задача *ZDT3*: исходный метод;  $\delta = 0,2$ ;  $\rho = 2$ ,  $\delta_{\min} = 0,001$ ;  
 $\hat{t} = 2000$

Лучший результат и соответствующие значения этих параметров иллюстрирует рисунок 8. Рисунок показывает, что метод даже для такой сложной задачи позволяет получить достаточно мощное ( $I_{ONVG}(\Theta) = 355$ ), плотное и равномерное покрытие фронта Парето. Для обеспечения столь высокого качества Парето-аппроксимации, потребовалось модификация исходной схемы метода, заключающаяся в разрешении использования крайних точек текущей аппроксимации  $\Theta$  в качестве центральных. Рисунок 7 иллюстрирует решение задачи без такой модификации.

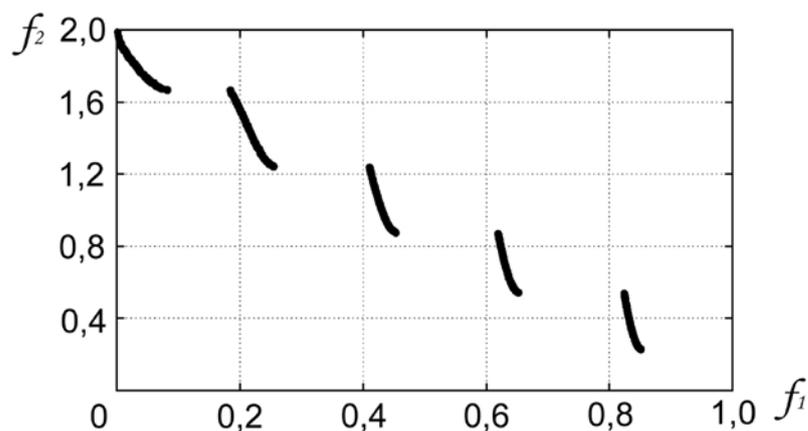


Рисунок 8 - Задача *ZDT3*: модифицированный метод;  $\delta = 0,2$ ;  $\rho = 2$ ,  
 $\delta_{\min} = 0,001$ ;  $\hat{t} = 2000$

*Задача ZDT6* (рисунки 9, 10) оказалась наиболее сложной для данного метода. Исследование выявило, что качество полученной Парето-аппроксимации в данном случае очень сильно зависит от значений свободных параметров метода, а также от начального приближения. Так при изменении коэффициента  $\rho$  с 1,5 до 1,6 (всего на 0,1!), метод порождает всего два решения ( $I_{ONVG}(\Theta) = 2$ ). Лучший результат, достигнутый методом, иллюстрирует рисунок 9 ( $I_{ONVG}(\Theta) = 86$ ). Ни равномерность, ни плотность полученной аппроксимации фронта Парето не являются приемлемыми.

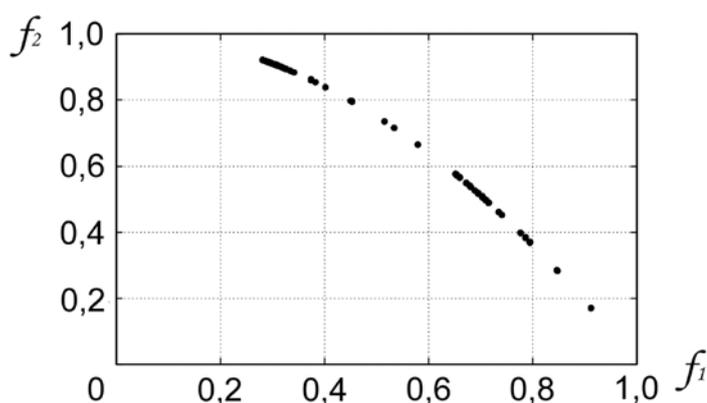


Рисунок 9 - Задача *ZDT6*:  $\delta = 1$ ;  $\rho = 1,5$ ,  $\delta_{\min} = 0,0001$ ;  $\hat{t} = 750$

На рисунке 10 представлены результаты решения той же задачи с помощью генетического алгоритма. Рисунок показывает, что генетический

алгоритм, в отличие от исследуемого метода AWS, не нашел ни одного решения, соответствующего условию  $f_1(X) < 0,4$ .

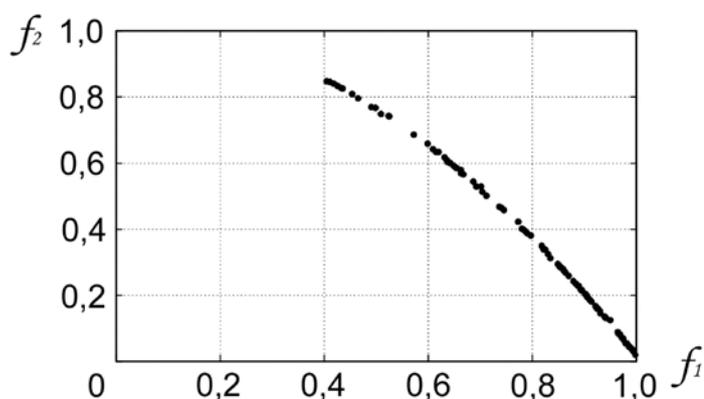


Рисунок 10 - Задача ZDT6: генетический алгоритм

Задача ZDT7 (рисунок 11). Метод в этом случае обеспечил высокое качество Парето-аппроксимации по критериям плотности и равномерности покрытия фронта Парето при достаточно мощном архивном множестве ( $I_{ONVG}(\Theta) = 511$ ) Исследование показало слабую зависимость качества аппроксимации от значений свободных параметров метода. Оказалось, что как и в задаче Аудета и по тем же причинам, некоторое число полученных решений не удовлетворяет ограничениям задачи.

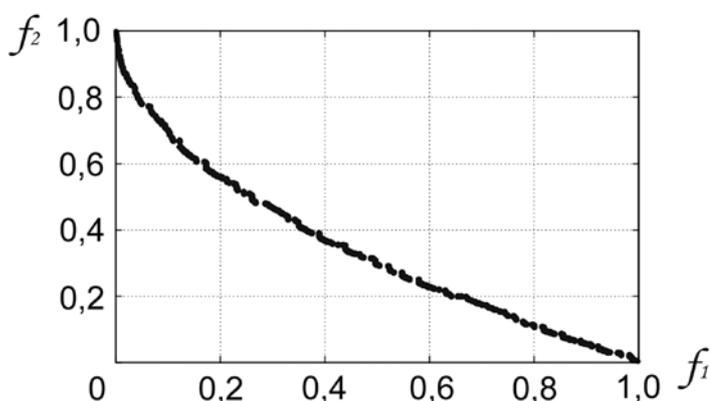


Рисунок 11 - Задача ZDT7:  $\delta = 0,5$ ;  $\rho = 1,25$ ,  $\delta_{\min} = 0,001$ ;  $\hat{t} = 5000$

## Заключение

Результаты исследования показывают, что метод AWS обеспечивает высокое качество Парето-аппроксимации в случае выпуклого, хотя, быть может, и несвязного фронта Парето. Для задач, имеющих вогнутый фронт Парето, метод не всегда обеспечивает удовлетворительное качество решения или обеспечивает его, но при значительном числе итераций. В некоторых случаях метод дает недопустимые решения, обусловленные используемым способом учета ограничений на область доверия. В целом, результаты исследования показывают высокий потенциал развития метода.

В продолжение работы авторы планируют исследовать альтернативные способы учета ограничений на область доверия, различные, в том числе нейросетевые, методы построения локальных метамоделей критериальных функций. Главной задачей является распространение метода на МКО-задачи с более, чем двумя критериями оптимальности.

Работа поддержана грантом РФФИ 12-07-00324-а.

## Литература

1. Ларичев О.И. Теория и методы принятия решений: Учебник для ВУЗов.- М.: Университетская книга, Логос, 2006.- 392 с.
2. Карпенко А.П., Митина Е.В., Семенихин А.С. Популяционные методы аппроксимации множества Парето в задаче многокритериальной оптимизации. Обзор // Наука и образование: электронное научно-техническое издание, 4, 2012, (<http://www.technomag.edu.ru/doc/363023.html>).
3. Jong-hyun Ryu, Sujin Kim, Hong Wan. Pareto front approximation with adaptive sum method in multiobjective simulation optimization // Proceedings of the 2009 Winter Simulation Conference, pp. 623 – 633.
4. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // Evolutionary Computation, 2000, Vol. 8(2), pp. 173-195.

5. Zitzler E., Thiele L., Marco Laumanns M., Fonseca C. M., da Fonseca V. G. Performance Assessment of Multiobjective Optimizers: An Analysis and Review // IEEE Transactions of Evolutionary Computation, 2003, Vol. 7(2), pp. 117-132.

6. Audet C., Savard G., Zghal W. Multiobjective optimization through a series of single-objective formulations // SIAM Journal on Optimization, 2006, 17(1), pp. 188–210.

**Adaptive weighted sum method for solving Pareto-approximation problem**

# 06, June 2012

DOI: **10.7463/0612.0423283**

Karpenko A.,P., Savelov A.S., Semenikhin A., S.

Russia, Bauman Moscow State Technical University

[apkarpenko@mail.ru](mailto:apkarpenko@mail.ru)[saspost@yandex.ru](mailto:saspost@yandex.ru)

The authors consider solving the discrete problem of Pareto set and frontier optimization in the multi-criterion optimization problem. The aim of this work was to examine the efficiency of Adaptive Weighted Sum method (AWS) which was initially developed by J-H. Ryu, S. Kim and H. Wan. Comparing to the original research, the authors used a wider range of multi-criterion benchmark optimization problems. A range of limitations of AWS methodology and usage complexity was found. Modification of the method was proposed to resolve the problems and limitations.

---

**Publications with keywords:** [multimodal optimization problem](#), [Pareto-set approximation](#), [adaptive weighted sum method](#)

**Publications with words:** [multimodal optimization problem](#), [Pareto-set approximation](#), [adaptive weighted sum method](#)

---

## References

1. Larichev O.I. *Teoriia i metody priniatiia reshenii* [Theory and methods of decision making]. Moscow, Universitetskaia kniga, Logos, 2006. 392 p.
2. Karpenko A.P., Mitina E.V., Semenikhin A.S. Populiatsionnye metody approksimatsii mnozhestva Pareto v zadache mnogokriterial'noi optimizatsii. Obzor [Review: population methods of Pareto set approximation in multi-objective optimization problem]. *Nauka i obrazovanie MGTU im. N.E. Baumana* [Science and Education of the Bauman MSTU], 2012, no. 4. Available at: <http://www.technomag.edu.ru/doc/363023.html> , accessed 08.07.2012.
3. Ryu J.-H., Kim S., Wan H. Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization. *Proc. of the 2009 Winter Simulation Conference (WSC)*, 2009, Austin, pp. 623-633. Available at: <http://www.informs-sim.org/wsc09papers/060.pdf> , accessed 08.07.2012.

4. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation (journal)*, 2000, vol. 8, no. 2, pp. 173-195.
5. Zitzler E., Thiele L., Marco Laumanns M., Fonseca C. M., da Fonseca V. G. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions of Evolutionary Computation*, 2003, vol. 7, no. 2, pp. 117-132.
6. Audet C., Savard G., Zghal W. Multiobjective optimization through a series of single-objective formulations. *SIAM Journal on Optimization*, 2008, vol. 19, no. 1, pp. 188–210.